Spatial Databases: Accomplishments and Research Needs

S. Shekhar, S. Chawla, S. Ravada
A. Fetterer, X. Liu, C.T. Lu
Computer Science Department, University of Minnesota
EE/CS 4-192, 200 Union St. SE., Minneapolis, MN 55455
[*shekhar*|*chawla*|*siva*|*fetterer*|*xliu*|*ctlu*]@cs.umn.edu
http://www.cs.umn.edu/research/shashi-group/

Abstract

Spatial databases have been an active area of research for over two decades, addressing the growing data management and analysis needs of spatial applications such as Geographic Information Systems. This research has produced a taxonomy of models for space, spatial data types and operators, spatial query languages and processing strategies, as well as spatial indexes and clustering techniques. However, more research is needed to improve support for network and field data, as well as query processing (e.g. cost models, bulk load). Another important need is to apply the spatial data management accomplishments to newer applications such as data warehouses and multimedia information systems. The objective of this paper is to identify recent accomplishments and the research needs in the near term.

Keywords: Spatial Databases, Multi-Dimensional, Object-Relational, Databases, Geographic Information Systems

1 Introduction

1.1 Spatial Databases

A spatial database [11, 15, 35] management system aims at the effective and efficient management of data related to a space such as the physical world (geography, urban planning, astronomy); parts of living organisms (anatomy of the human body); engineering design (very large scale integrated circuits, the design of an automobile or the molecular structure of a pharmaceutical drug); and conceptual information space (a multi-dimensional decision support system, fluid flow, or an electro-magnetic field).

The field of spatial database research has been an active area of research for over two decades. The results of this research, e.g. spatial multi-dimensional indexes, are being used in a number of areas. The field of spatial databases can be defined by its accomplishments; current research is aimed at improving its functionality and its performance. The impetus for improving functionality comes from the needs of existing applications such as Geographic Information Systems (GIS) and Computer Aided Design (CAD), as well as from potential applications such as Multimedia Information System(MMIS), Data Warehousing (DWH) and NASA's Earth Observation System (EOS). The acceptance of GIS as an important tool in government decision-making is also documented [34] and military planners have embraced GIS technology at all levels of tactical, operational and strategic planning, including battlefied visualization and terrain analysis [20].

Commercial examples of spatial database management include Informix's spatial datablades (i.e. 2D, 3D, Geodetic), Oracle's Universal server with either Spatial Data Option or Spatial Data Cartridge and ESRI's Spatial Data Engine (SDE). Research prototype examples of spatial database management systems include spatial datablades with Postgres [30], GeO2, and Paradise [9]. The functionalities provided by these systems include a set of spatial data types such as a point, line-segment and polygon, and a set of spatial operations such as inside, intersection, and distance. The spatial types and operations may be made a part of a query language such as SQL, which allows spatial querying when combined with an object-relational database management system [6, 32]. The performance enhancement provided by these systems includes a multi-dimensional spatial index and algorithms for spatial access methods, spatial range queries and spatial joins. Spatial indexing with concurrency control may be implemented in the object-relational server for performance reasons.

Existing and emerging applications require new functionalities including the modeling of network spaces and continuous fields. The performance needs of emerging applications require not only the management of large data-sets, but also new processing strategies for spatial set-operations, field operations (e.g. slope), and network analysis (e.g. shortest-path, route-evaluation).

1.2 Related Work and Our Contributions

Recent reports[11, 15, 35, 1] have described the accomplishments of spatial database research and have prioritized research needs. A broad survey of spatial database requirements and an overview of research results is provided by [35, 11, 1]. Basic modeling requirements for spatial objects such as points, lines, and polygons are given in terms of their geometry, topology and object relationships (topological, directional, metric, network). Requirements are given for other user-level issues such as graphical input and output and query language support. Spatial clustering and indexing techniques [23] such as Grid-files, Z-order, Quad-tree, Kd-trees, R-trees [12] and associated join strategies are described. Finally, an architecture for spatial databases is given in terms of the object-relational model.

Research needed to improve the performance of spatial databases in the context of object-relational databases was listed in [15]. The primary research needs identified were concurrency control techniques for spatial indexing methods, the development of cost models for query strategies, and the development of new spatial join algorithms beyond nested-loop and tree matching.

Many of the research needs identified in [15] have since been addressed. For example, concurrency control techniques for R-trees have been studied in the context of R-link [16] trees. Also, new spatial join strategies using space partitioning [22] have been explored. In this paper, we identify the recent accomplishments in spatial databases as well as current research needs, based on publications in journals and conference proceedings and recent commercial trends.

1.3 Scope and Outline

The role of the spatial database component is dependent on the type of database management system (DBMS) involved: relational, object-oriented or object-relational. In this paper, we focus the discussion of spatial databases in the context of the object-relational [6, 32, 31] databases, which provide extensibility to many components of traditional databases to support new application domains. These and other important issues including architectural options, Raster DBMS and Network spaces are covered in detail in our forthcoming book [24]. Spatial databases have been one of the most common applications of object-relational databases and have influenced their design a great deal. Object-relational databases allow the inclusion of spatial data-types, spatial operations, and multi-dimensional indexing systems. This three-layer architectural framework is shown in Figure 1, and it consists of an object-relational database management system, a spatial database, and a spatial application such as a GIS or MMIS. The interface between the application and the spatial data system maps application-specific constructs to the spatial database. The spatial database associates the application requirements to the functionality provided by the DBMS. The interface to the DBMS supports specialized query processing, which in turn supports the core database requirements for achieving acceptable performance.

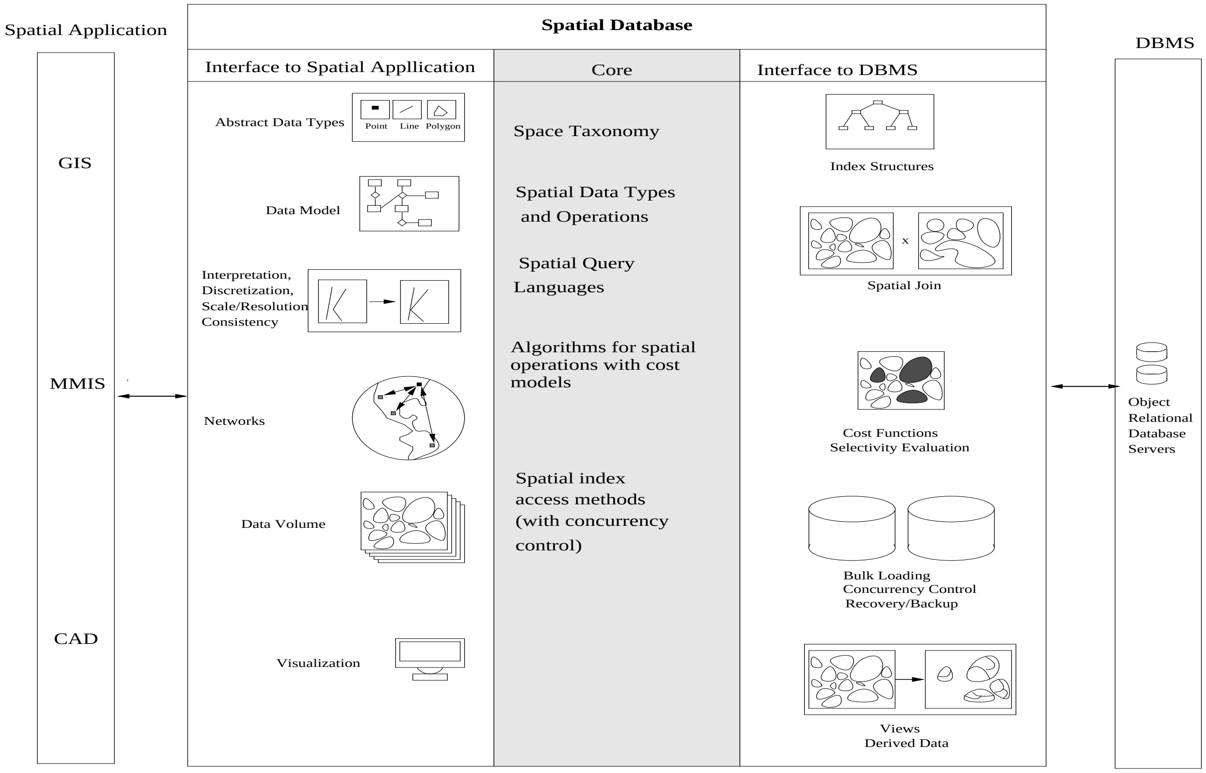


Figure 1: 3-layer architecture

Emerging trends such as world-wide-web interfaces, multimedia data, and image processing are likely to impact the data sharing and analysis needs of spatial databases. Scaling up to large datasets requires new research in many areas beyond spatial databases, including research on file-systems, device-drivers for tertiary storage, computer networks, and visualization software and algorithms related to graphics and computational geometry. This paper does not explore those issues.

The remainder of the paper is organized as follows: Section 2 describes the recent advances in spatial databases. Section 3 states the research needs for spatial databases. Section 4 highlights our conclusions and motivates exploration of applications whose needs are not currently met by spatial databases.

2 Accomplishments

Research into spatial databases has mainly focused on developing a space taxonomy, spatial data models, spatial query languages and processing strategies, and spatial access methods. This section lists recent important accomplishments, not only for the current applications of spatial databases, but also for the emerging database problems that have spatial dimensions.

2.1 Space Taxonomy

Space is a framework to formalize specific relationships among a set of objects. Depending on the relationships of interest, different models of space such as set-based space, topological space, euclidean space, metric space and network space can be used [35]. Set-based space uses the basic notion of elements, element-equality, sets and membership to formalize the set relationships such as set-equality, subset, union, cardinality, relation, function, and convexity. Relational and object-relational databases use this model of space.

Topological space uses the basic notion of a neighborhood and points to formalize the extended object relationships such as boundary, interior, open, closed, within, connected, and overlaps, which are invariant under elastic deformation. Combinatorial topological space formalizes relationships such as Euler's formula (#faces + #vertices - #edges =1 for planar configuration). Network space is a form of topological space in which the connectivity property among nodes formalizes graph properties such as connectivity, iso-morphism, shortest-path, and planarity.

Euclidean coordinatized space uses the notion of a coordinate system to transform spatial properties and relationships to properties of tuples of real numbers. Metric spaces formalize the distance relationships using positive symmetric functions that obey the triangle inequality. Many multidimensional applications use euclidean coordinatized space with metrics such as distance.

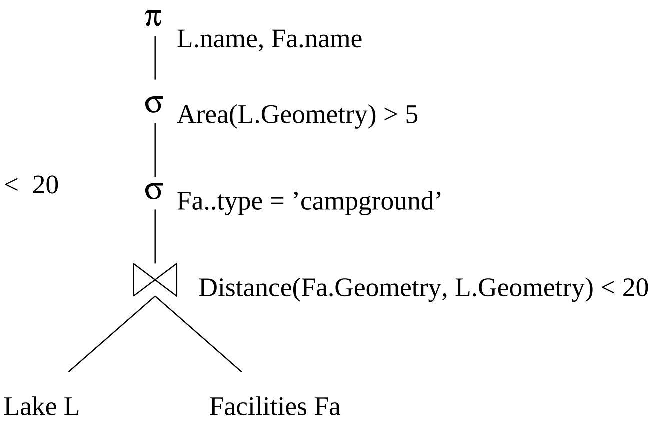
2.2 Spatial Data Model and Query Language

A spatial data model [25, 35] is a type of data-abstraction that hides the details of data-storage. There are two common models of spatial information: field-based and object-based. The field-based model treats spatial information such as altitude, rainfall and temperature as a collection of spatial functions transforming a space-partition to an attribute domain. The object-based model treats the information space as if it is populated by discrete, identifiable, spatially-referenced entities. The operations on spatial objects include *distance* and *boundary*. The operations on fields include local, focal, and zonal operations, as shown in Table 2. The fields may be continuous, differentiable, discrete, and isotropic or anisotropic, with positive or negative auto-correlation. Certain field operations (slope or interpolation) assume certain field properties (differentiable or positive auto-correlation).

An implementation of a spatial data model in the context of object-relational databases consists of a set of spatial data types and the operations on those types. Much work has been done over the last decade on the design of spatial Abstract Data Types(ADTs) and their embedding in a query language. Consensus is slowly emerging via standardization efforts, and recently the OGIS consortium [21] has proposed a specification for incorporating 2D geospatial ADTs in SQL. Figure 3, which illustrates this spatial data-type hierarchy consists of *Point, Curve* and *Polygon* classes and a parallel class of *Geometry Collection.* The basic operations operative on *all* datatypes are shown in Table 1. The topological operations are based on the ubiquitous 9-intersection model [10]. Using the OGIS specification, common spatial queries can be intutively posed in SQL. For example, the query *Find all lakes which have an area greater than 5 sq. km. and are within 20 km. from the campgrounds* can be posed as shown in Figure 2(a).

```
SELECT   L.name, Fa.name
FROM     Lake L, Facilities Fa
WHERE    Area(L.Geometry) > 5 AND
         Fa.type = ’campground’  AND
         Distance(Fa.Geometry, L.Geometry) <  20
```

(a)



(b)

Figure 2: (a) SQL query with spatial operators. (b) Corresponding query tree.

Other example GIS queries which can be implemented using OGIS operations are provided in Table 3. The OGIS specification is confined to topological and metric operations on vector data types. Other interesting classes of operations are network, direction, dynamic and the field operations of focal, local and zonal(see Table 2). While standards for field based raster data types are still emerging, Map Algebra [33], specifically designed for cartographic modeling and RaSQL, based on Image Algebra [3], for general multi-dimensional discrete objects(satellite images, X-rays, etc.), are important milestones.

2.3 Spatial Query Processing

The efficient processing of spatial queries requires both efficient representation and efficient algorithms. Common representations of spatial data in an object model include spaghetti, the node-arc-area(NAA) model, the doubly-connected-edge-list (DCEL), and boundary representation [17], some of which are shown in Figure 4 using entity-relationship diagrams. The NAA

Basic Functions	SpatialReference()	Returns the Reference System of the geometry
	Envelope()	The minimum bounding rectangle of the geometry
	Export()	Convert the geometry into a different representation.
	IsEmpty()	Tests if the geometry is a empty set or not.
	IsSimple()	Returns True if the geometry is simple(no self-intersection)
	Boundary()	Returns the boundary of the geometry
Topological/ Set Operators	Equal	Tests if the geometries are spatially equal
	Disjoint	Tests if the geometries are disjoint.
	Intersect	Tests if the geometries intersect
	Touch	Tests if the geometries touch each other.
	Cross	Tests if the geometries cross each other.
	Within	Tests if the given geomtry is within another given geometry
	Contains	Tests if the given geometry contains another given geometry
	Overlap	Tests if the geometry overlaps another geometry.
Spatial Analysis	Distance	Returns the shortest distance between two geometries
	Buffer	Returns a geometry that represents all points whose distance from the given is less than or equal to the specified distance
	ConvexHull	Returns the convex hull of the geometry
	Intersection	Returns the intersection of two geometries
	Union	Returns the union of two geometries
	Difference	Returns the difference of two geometries
	SymDiff	Returns the symmetric difference of two geometries

Table 1: Representative functions specified by OGIS[21]

model differentiates between the topological concepts (node, arc, areas) and the embedding space (points, lines, areas). The spaghetti-ring and DCEL focus on the topological concepts. The representation of the field data model includes a regular tessellation (triangular, square, hexagonal grid), as well as triangular irregular networks (TIN).

The spatial queries[7] shown in Table 3 are often processed using *filter* and *refine* techniques. Approximate geometry such as the minimal orthogonal bounding rectangle of an extended spatial object is first used to filter out many irrelevant objects quickly. Exact geometry is then used for the remaining spatial objects to complete the processing. Strategies for range-queries include a scan and index-search in conjunction with the plane-sweep algorithm [5]. Strategies for the spatial-join include the nested loop, tree matching [5] when indices are present on all participating relations, and space partitioning [22] in the absence of indices. To speed up computation for large spatial objects (it is common for polygons to have 1000 or more edges), object indices are used in extended filtering. Strategies such as object approximation and tree matching originated in spatial-databases, and can potentially be applied in other domains with similar characteristics.

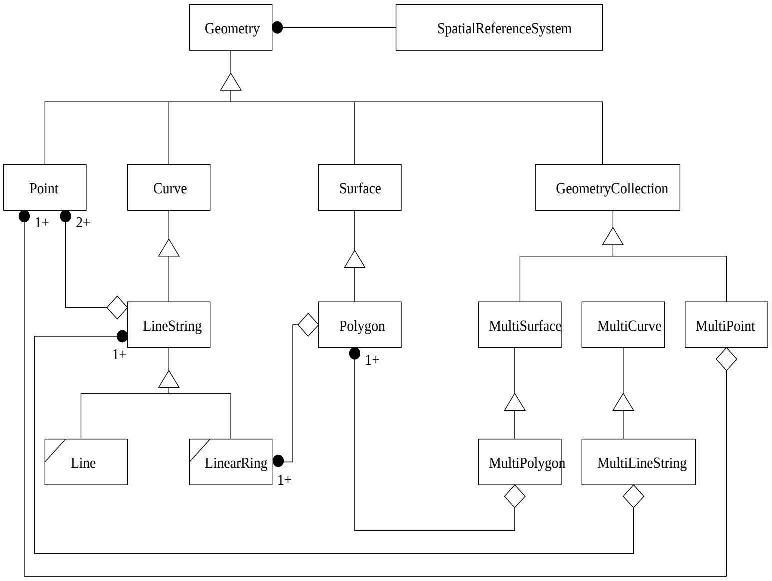


Figure 3: Spatial Data Type Hierarchy[21]

2.4 Spatial File Organization and Indices

The physical design of a spatial database optimizes the instructions to storage devices for performing common operations on spatial data files. File designs for secondary storage include clustering methods as well as spatial hashing methods. The design of spatial clustering techniques is more difficult compared to the design of traditional clustering because there is no natural order in multidimensional space where spatial data resides. This is only complicated by the fact that the storage disk is a logical one-dimensional device. Thus, what is needed is a mapping from a higher dimensional space to a one-dimensional space which is *distance-preserving*: so that elements that are close in space are mapped onto nearby points on the line, and one-one: no two points in the space are mapped onto the same point on the line [2]. Several mappings, none of them ideal, have been proposed to accomplish this. The most prominent ones include *row-order, z-order* and the *Hilbert-curve*(Figure 5).

Metric clustering techniques use the notion of distance to group nearest neighbors together in a metric space. Topological clustering methods like connectivity-clustered access methods [27] use the min-cut partitioning of a graph representation to efficiently support graph traversal operations. The physical organization of files can be supplemented with indices, which are data-structures to improve the performance of search operations.

Classical one-dimensional indices such as the B^+ tree can be used for spatial data by linearizing a multi-dimensional space using a space-filling curve such as the Z-order(see Figure 5). A large number of spatial indices [23] have been explored for multi-dimensional euclidean space. Representative indices for point objects include Grid files, multi-dimensional grid files

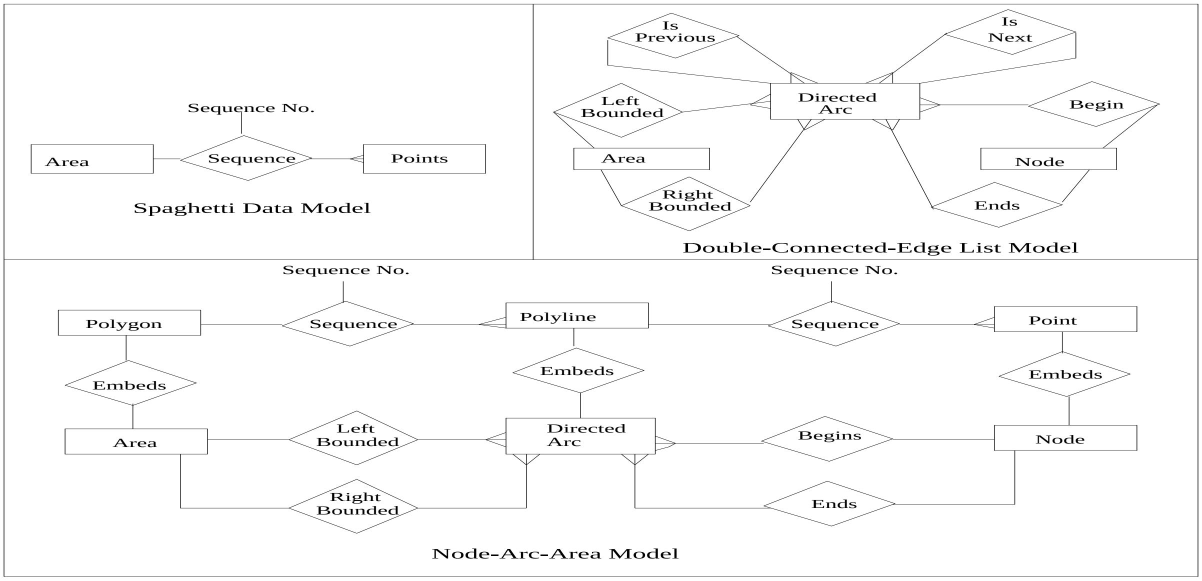


Figure 4: Entity Relationship Diagrams for Common Representations of Spatial Data

[18], Point-Quad-Trees, and Kd-trees. Representative indices for extended objects include the R-tree family, the Field tree, Cell tree, BSP tree, and Balanced and Nested grid files.

One of the first access methods created to handle extended objects was Guttman's R-tree structure [12]. The R-tree is a height balanced natural extension of the B+ tree for higher dimensions. Objects are represented in the R-tree by their minimum bounding rectangles(MBRs). Nonleaf nodes are composed of entries of the form $(R, child - pointer)$, where R is the MBR of all entries contained in the *child-pointer*. Leaf nodes contain the MBRs of the data objects. To guarantee good space utilization and height-balance, the parent MBRs are allowed to overlap. Figure 6(a) illustrates the spatial objects organized in an R-tree, while Figure 6(b) shows the

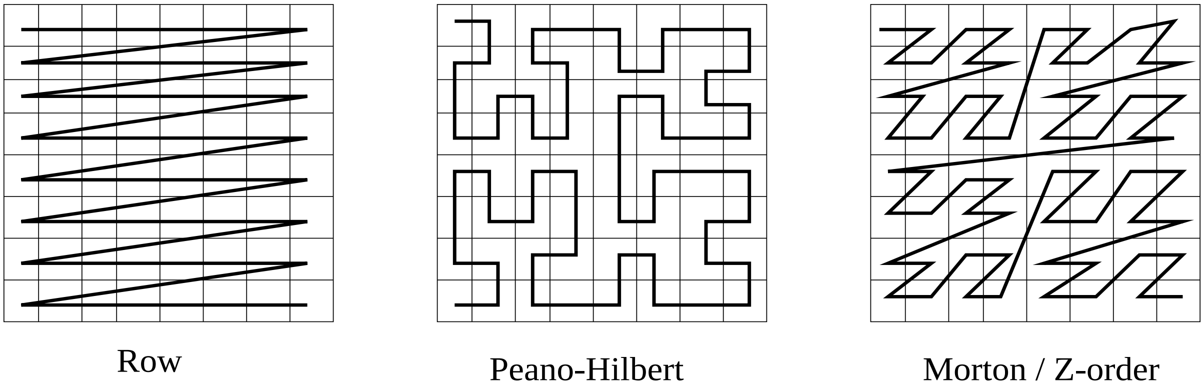


Figure 5: Space-Filling Curves to Linearize a Multidimensional Space

Data model	Operator Group	Operation
Vector Object	Set-Oriented	equals, is a member of, is empty, is a subset of, is disjoint from, intersection, union, difference, cardinality
	Topological	boundary, interior, closure, meets, overlaps, is inside, covers, connected, components, extremes, is within
	Metric	distance, bearing/angle, length, area, perimeter.
	Direction	east, north, left, above, between.
	Network	successors, ancestors, connected, shortest-path
	Dynamic	translate, rotate, scale, shear, split, merge
Raster field	Local	Point-wise sums, differences, maximums, means, etc
	Focal	slope, aspect, weighted average of neighborhood
	Zonal	sum or mean or maximum of field values in each zone

Table 2: A Sample of Spatial Operations

file structure where the nodes correspond to disk pages. Many variations of the R-tree structure exist whose main emphasis is on discovering new strategies to maintain the balance of the tree, in case of a split, and to minimize the overlap of the MBRs in order to improve the search time.

Concurrency control for spatial access methods [16] is provided by the R-link tree, which is a variant of the R-tree with additional sibling pointers that allow the tracking of modifications. Concurrency is provided during operations such as search, insert, and delete. The R-link tree is also recoverable in a write-ahead logging environment.

Single Table Queries	
Grouping	Recode all land with silty soil to silt-loam soil
Isolate	Select all land owned by Steve Steiner
Classify	If the population density is less than 100 people / sq. mi., land is acceptable
Scale	Change all measurement to the metric system
Rank	If the road is an Interstate, assign it code 1; if the road is a state or US Highway, assign it code 2; otherwise assign it code 3
Evaluate	If the road code is 1, then assign it Interstate; if the road code is 2, then assign it Main Artery; if the road code is 3, assign it Local Road
Rescale	Apply a function to the population density
Multi-Table Queries	
Attribute Join	Join the Forest layer with the layer containing forest-cover codes
Zonal	Produce a new map showing state populations given county population
Registration	Align two layers to a common grid reference
Spatial Join	Overlay the land-use and vegetation layers to produce a new layer

Table 3: Typical Spatial Queries from GIS

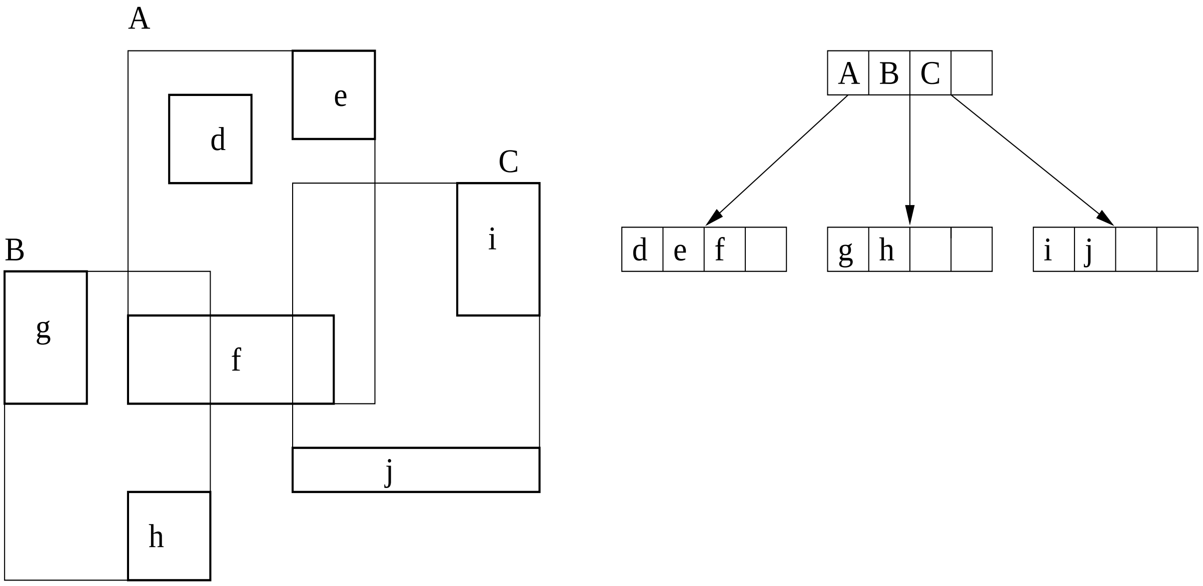


Figure 6: (a) Spatial objects(bold) arranged in R-tree hierarchy, (b) R-tree file structure on disk

2.5 Other Accomplishments

Spatial applications like NASA's Earth Observation System(EOS) have some of the largest data sets encountered in any application to date. This has prompted new research in database-file design for storage on tertiary storage devices such as juke-boxes. Representative results include those from the Sequoia 2000 project[30]. High-performance spatial applications such as flight simulators with geographic accuracy have triggered the development of new parallel formalizations for the range query and the spatial join query, including declustering methods and dynamic-load balancing techniques for multi-dimensional spatial data[28, 19]. Other interesting developments include hierarchical algorithms for shortest path computation [14] and view materialization [26].

3 Research Needs

Spatial databases are being used for an increasing number of new applications, such as Intelligent Transportation Systems, NASA's Earth Observation System, Multimedia Information Systems (MMIS) and Data Warehouses. This section lists representative research needs.

3.1 Space Taxonomy

Many spatial applications manipulate continuous spaces of different scales and with different levels of discretization. A sequence of operations on discretized data can lead to growing errors similar to the ones introduced by finite-precision arithmetic on numbers. There are preliminary results [11] on the use of discrete basis and bounding errors with peg-board semantics. Another related problem concerns interpolation to estimate the continuous field from a discretization. Negative spatial auto-correlation makes interpolation error-prone. Further work is needed on a framework to formalize the discretization process, its associated errors, and on interpolation.

3.2 Spatial Data Model

Spatial data models have been developed for topological, metric and coordinatized euclidean space. The OGIS specification alluded to in Section 2.2 is confined to topological operators [8] and more work is needed to incorporate relationships which involve directional [29] and metric properties (see Table 2 for examples). In addition there has been very little work towards developing data models, data types (e.g. node, edge, path), and a kernel set of operations (e.g. get-successors, shortest path) for network space, despite their critical role in applications like transportation and utility management (telephone, gas, electric).

Similarly, there is a need for developing the field data model [33] towards a field-based query language. Operations on fields will be needed to help derive new information such as land-cover classification; the fields involved include temperature, texture, and water content, and are obtained through imaging in different bands such as other infrared, visible bands, or microwave.

3.3 Spatial Query Processing

Many open research areas exist at the logical level of query processing, including query-cost modeling and strategies for nearest neighbor, bulk loading as well as queries related to fields and networks. Cost models are used to rank and select the promising processing strategies, given a spatial query and a spatial data set. Traditional cost models may not be accurate in estimating the cost of strategies for spatial operations, due to the distance metric as well as the semantic gap between relational operators and spatial operation. Cost models are needed to estimate the selectivity of spatial search and join operations towards comparison of execution-costs of alternative processing strategies for spatial operations during query optimization. Preliminary work in the context of the R-tree, tree-matching join, and fractal-models is promising [4, 36] , but more work is needed.

Similarly, common strategies employed in traditional databases for the *logical transformation* step in query optimization may not be always applicable in the context of spatial databases.

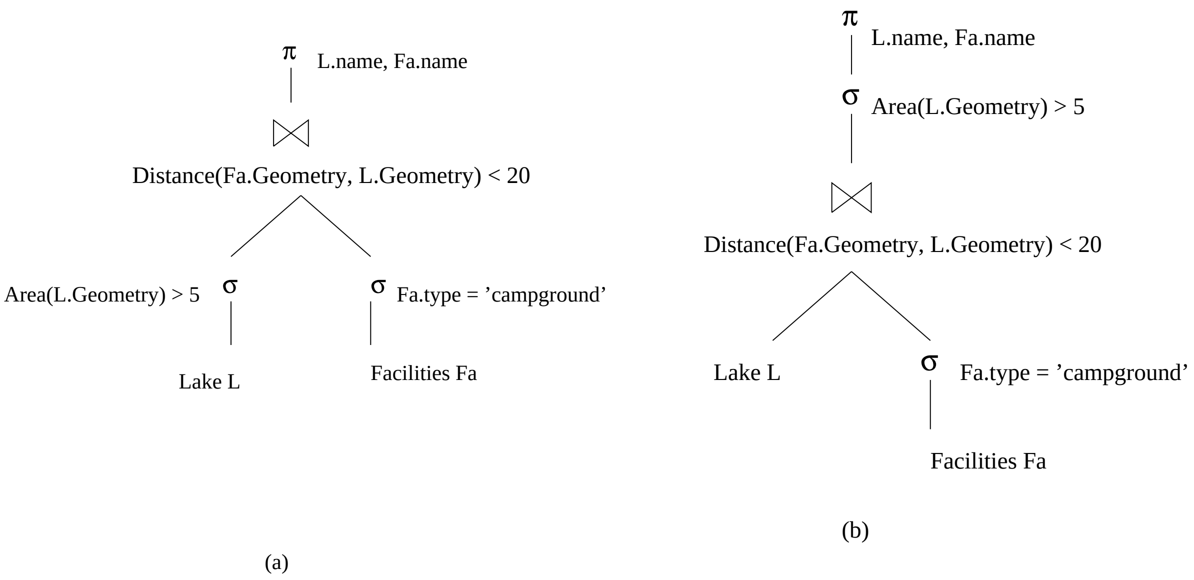


Figure 7: (a): Area() before Distance(). (b): Distance() before Area().

For example consider the query(see Figure 2(a)) first introduced in Section 2. Let us assume that the **Area()** function is not pre-computed and that its value is computed afresh every time it is invoked. A query tree generated for the query is shown in Figure 2(b).

In the classical situation, the rule “select before join” would dictate that the **Area()** function be computed before the join predicate function, **Distance()**(Figure 7(a)), the underlying assumption being that the computational cost of executing the select and join predicate are equivalent and negligible compared to the I/O cost of the operations. In the spatial situation the relative cost per tuple of **Area()** and **Distance()** is an important factor in deciding the order of the operations [13]. Depending upon the implementation of these two functions the optimal strategy may be to process the join before the select operation(see Figure 7(b)).

Many processing strategies using the overlap predicate have been developed for range queries and spatial join queries. However, there is a need to develop and evaluate strategies for many other frequent queries such as those in Table 4. These include queries on objects using predicates other than overlap and queries on fields such as slope analysis as well as queries on networks such as the shortest path to a set of destinations. Bulk loading strategies for spatial data also need further study.

3.4 Spatial File Organization and Indices: Physical Level

Many file organizations and indices with distance metrics have been developed for coordinatized euclidean space. However, little work has been done on file clustering and on indices for network spaces such as road-maps and telephone networks. Further work is needed, both to characterize

Buffer	Find the areas 500 ft. from power lines
Voronoize	Classify households as to which supermarket they are closest to
Neighborhood	Determine slope based on elevation
Network	Find the shortest path from the warehouse to all delivery stops
Allocation	Where is the best place to build a new restaurant
Transformation	Triangulate a layer based on elevation
Bulk Load	Load a spatial data file into the database
Raster ↔ Vector	Convert between raster and vector representations

Table 4: Difficult Spatial Queries from GIS

the access patterns of the graph algorithms that underlie network operations and to design access methods.

The R-link tree [16] is among the few approaches available for concurrency control on the R-tree. New approaches for concurrency-control techniques are needed for other spatial indices. The data volume of emerging spatial applications such as NASA's EOS is among the highest of any database application. Sequoia 2000 [30] provides an approach towards tertiary storage files and indices. Other approaches for managing databases on tertiary storage need to be investigated.

3.5 Other

Other research needs include benchmarking, work-flow modeling, and the visual presentation of results. The Sequoia 2000 [30] benchmark characterizes the data and queries in Earth Science applications. The performance of loading data, raster queries, spatial selection, spatial joins, and recursion is addressed in 11 benchmark queries. A few more are provided in the Paradise system [9]. Similar benchmarks are needed to characterize the spatial data management needs of other applications such as GIS, DWH, and transportation.

The work-flow in some spatial applications such as GIS is based on manipulating layers to produce new, derived layers. Typically, the layers are combined in a tree-based manner, starting with a large number of source layers and producing new layers until a final result layer is produced. Information about dependence among layers is useful for change propagation if the source layers are modified.

Spatial databases may require a different type of concurrency support than is needed by traditional databases. For example, transactions in traditional systems tend to be short (on the order of seconds). However, in spatial databases, these transactions can last up to a couple of hours for editing and browsing. Similarly, recovery and backup issues may also change, as the spatial objects tend be large (a few megabytes) when compared to their counterparts in traditional systems. There is a need to characterize the work flow of spatial applications.

Many spatial applications present results visually, in the form of maps which consist of graphic images, 3D displays, and animations. They also allow users to query the visual representation by pointing to the visual representation using devices like a mouse or a pen. Further work is needed to explore the impact of querying by pointing and visual presentation of results on database performance.

4 Summary and Discussion

In this survey we have presented the major research accomplishments and techniques which have emerged from the area of SDBMS. These include object-based data modeling, spatial data types, filter and refine techniques for query processing and spatial indexing. We have also identified areas where more research is needed. Some of these areas are spatial graphs, field based modeling, cost modeling and concurrency control, query processing techniques and discretization and propogation error.

Many of the spatial techniques highlighted in this survey are being used in an increasing number of applications such as GIS, CAD, and EOS. We believe that other emerging multi-dimensional applications such as multimedia information systems will use these methods to solve problems such as searching and indexing spatial content. We illustrate the possibilities in the context of multi-media information systems with text, audio and video data over the world-wide-web.

Multimedia data has a spatial content which can be queried using the same spatial operators that have become popular in geographic information systems. For example, the spatial operator *inside of* can be applied to text to locate sentences that contain the word “multimedia”. Also, audio is often broken into channels with each channel containing input from a different source; for instance, trumpet, guitar, and voice. These channels are analogous to layers in GIS and can be manipulated similarly. A spatial join could determine all of the locations where the input from both piano and voice is over a certain decibel threshold.

A video database such as a movie server can take advantage of techniques developed for spatial databases. Consider the movie *Toy Story*: each frame contains spatial content with objects interacting in topological relationships. For instance, Buzz Lightyear could be above the trees when he is flying, and frames in the movie could be queried based on those relationships. For example, if you cannot remember when in the movie an important event occurred, but you can remember that Buzz Lightyear was in front of a tree, you would be able to query the movie using that relationship to determine when in the movie that event took place. Such queries exploit the topological relationships inherent in all tangible objects.

Acknowledgments

This work is sponsored in part by the Army High Performance Computing Research Center under the auspices of the Department of the Army, Army Research Laboratory cooperative agreement number DAAH04-95-2-0003/contract number DAAH04-95-C-0008, the content of which does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred. This work was also supported in part by NSF grant #9631539 Thanks to Professor Jaideep Srivastava for technical commentary and to Christiane McCarthy for helping to improve the readability of the paper.

References

[1] N. Adam and A. Gangopadhyay. *Database issues in Geographical Information Systems.* Kluwer Academics, 1997.

[2] T. Asano, D. Ranjan, T. Roos, E. Wiezl, and P. Widmayer. Space filling curves and their use in the design of geometric data structures. *Theoretical Computer Science*, 181(1):3–15, July 1996.

[3] P. Baumann. Management of multidimensional discrete data. *VLDB Journal, Special issue on Spatial Database Systems*, 3(4):401–444, October 1994.

[4] A. Belussi and C. Faloutsos. Estimating the Selectivity of Spatial Queries Using the 'Correlation' Fractal Dimension. In *Proceedings of 21st International Conference on Very Large Data Bases(VLDB'95)*, pages 299–310, Zurich, Switzerland, September 1995.

[5] Thomas Brinkhoff, Hans-Peter Kriegel, and Bernhard Seeger. Efficient processing of spatial joins using R-trees. In *Proceedings of the 1993 ACM-SIGMOD Conference on the Management of Data*, pages 237–246, Washington D.C., June 1993.

[6] D. Chamberlin. *Using the New DB2: IBM's Object Relational System.* Morgan Kaufmann, 1997.

[7] N. Chrisman. *Exploring Geographic Information Systems.* John Wiley and Sons, 1997.

[8] E. Clemintini and P. Di Felice. Topological invariants for lines. *IEEE Transactions on Knowledge and Data Engineering*, 10(1):38–54, 1998.

[9] David J. DeWitt, Navin Kabra, Jun Luo, Jignesh M. Patel, and Jie-Bing Yu. Client-Server Paradise. In *Proceedings of the 20th International Conference on Very Large Data Bases,(VLDB'94)*, pages 558–569, Santiago de Chile, Chile, September 1994.

[10] M. Egenhofer. Spatial SQL: A Query and Presentation Language. *IEEE Transactions on Knowledge and Data Engineering*, 6(1):86–95, 1994.

[11] R.H. Güting. An Introduction to Spatial Database Systems. *VLDB Journal, Special issue on Spatial Database Systems*, 3(4):357–399, 1994.

[12] R. Guttman. R-tree: A dynamic index structure for spatial searching. In *Proceedings of the ACM SIGMOD Conference, Annual Meeting*, pages 47–57, Boston, MA., 1984.

[13] J.M. Hellerstein and M. Stonebraker. Predicate Migration:Optimizing Queries with Expensive Predicates. In *Proceedings of the ACM-SIGMOD International Conference on Management of Data*, pages 267–276, Washington, D.C, May 1993.

[14] N. Jing, Y. Huang, and E. Rundensteiner. Hierarchical encoded path views for path query processing: An optimal model and its performance evaluation. *IEEE Transactions on Knowledge and Data Engineering*, 10(3):409–432, 1998.

[15] W. Kim, J. Garza, and A. Kesin. Spatial Data Management in Database Systems. In *Advances in Spatial Databases, 3rd International Symposium, SSD'93 Proceedings , Lecture notes in Computer Science, Vol. 692, Springer, ISBN 3-540-56869-7*, pages 1–13, Singapore, 1993.

[16] M. Kornacker and D. Banks. High-Concurrency Locking in R-Trees. In *Proceedings of 21st International Conference on Very Large Data Bases(VLDB'95)*, pages 134–145, Zurich, Switzerland, September 1995.

[17] R. Laurini and D. Thompson. *Fundamentals of Spatial Information Systems.* Academic Press, 1992.

[18] J. Lee, Y. Lee, K. Whang, and I. Song. A physical database design method for multidimensional file organization. *Information Sciences*, 120(1):31–65, October 1997.

[19] D-R. Liu and S. Shekhar. A Similarity Graph-Based Approach to Declustering Problems and Its Application Toward Parallelizing Grid Files. In *Proceedings of the 11th International Conference on Data Engineering*, pages 373–381, Taipei, Taiwan, March 1995.

[20] US Army Corps of Engineers. Topographic engineering center. http://www.tec.army.mil/gis-internet2.html.

[21] Open GIS Consortium, Inc., http://www.opengis.org/public/abstract.html. *OpenGIS Simple Features Specification For SQL*, 1998.

[22] Jignesh M. Patel and David J. DeWitt. Partition Based Spatial-Merge Join. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 259–270, Montreal, Quebec, CA., 1996.

[23] H. Samet. *The Design and Analysis of Spatial Data Structures.* Addison-Wesley, 1990.

[24] S. Shekhar and S. Chawla. *Spatial Databases: Concepts, Implementation and Trends.* First draft, http://www.cs.umn.edu/Research/shashi-group/Book/index.html, 1998.

[25] S. Shekhar, M. Coyle, D-R. Liu, B. Goyal, and S. Sarkar. Data Models in Geographic Information Systems. *Communication of the ACM*, 40(4):103–111, 1997.

[26] S. Shekhar, Andrew Fetterer, and Brajesh Goyal. Materialization Trade-Offs in Hierarchical Shortest Path Algorithms. In *Advances in Spatial Databases, 5th International Symposium, SSD'97, Proceedings. Lecture Notes in Computer Science, Vol. 1262, Springer, ISBN 3-540-632238-7,* pages 94–111, 1997.

[27] S. Shekhar and D-R. Liu. A Connectivity-Clustered Access Method for Networks and Network Computation. *IEEE Transactions on Knowledge and Data Engineering*, 9(1):102–119, January 1997.

[28] S. Shekhar, S. Ravada, V. Kumar, D. Chubb, and G. Turner. Parallelizing a GIS on a Shared Address Space Architecture. *IEEE Computer*, 29(12), December 1996.

[29] Shashi Shekhar and X. Liu. Direction as a spatial object. In *ACM GIS Workshop[Accepted]*, Maryland, November 1998. Also available at http://www.cs.umn.edu/Research/shashi-group/paper_list.html.

[30] M. Stonebraker, J. Frew, and J. Dozier. The Sequouia 2000 Storage Benchmark. In *Proceedings of ACM SIGMOD Conference on the Management of Data*, pages 2–11, Washington D.C., May 1993.

[31] M. Stonebraker and G. Kennitz. POSTGRES Next-Generation Database Management System. *Communication of the ACM*, 34(10):78–92, 1993.

[32] M. Stonebreaker and D. Moore. *Object Relational DBMSs: The Next Great Wave.* Morgan Kaufmann, 1997.

[33] C.D. Tomlin. *Geographic Information systems and Cartographic Modeling.* Englewood Cliffs, NJ:Prentice-Hall, 1990.

[34] UCGIS. Ucgis congressional breakfast. http://urban.rutgers.edu/ucgis, 1998.

[35] M.F. Worboys. *Geographic Information Systems: A Computing Perspective.* Taylor and Francis, 1995.

[36] Y.Theodoridis, E. Stefanakis, and T. Sellis. Cost models for join queries in spatial databases. In *Proceedings of the 14th International Conference on Data Engineering*, pages 476–483, Orlando, Florida, Feb 1998.